METHOD AND SYSTEM FOR TERMINATING WRITE COMMANDS IN A HUB-BASED MEMORY SYSTEM

TECHNICAL FIELD

This invention relates to computer systems, and, more particularly, to a computer system including a system memory having a memory hub architecture.

BACKGROUND OF THE INVENTION

Computer systems use memory devices, such as dynamic random access memory ("DRAM") devices, to store data that are accessed by a processor. These memory devices are normally used as system memory in a computer system. In a typical computer system, the processor communicates with the system memory through a processor bus and a memory controller. The processor issues a memory request, which includes a memory command, such as a read command, and an address designating the location from which data or instructions are to be read. The memory controller uses the command and address to generate appropriate command signals as well as row and column addresses, which are applied to the system memory. In response to the commands and addresses, data are transferred between the system memory and the processor. The memory controller is often part of a system controller known as a "north bridge," which also includes bus bridge circuitry for coupling the processor bus to an expansion bus, such as a peripheral connect interface ("PCI") bus.

Although the operating speed of memory devices has continuously increased, this increase in operating speed has not kept pace with increases in the operating speed of processors. Even slower has been the increase in operating speed of memory controllers coupling processors to memory devices. The relatively slow speed of memory controllers and memory devices limits the data bandwidth between the processor and the memory devices.

In addition to the limited bandwidth between processors and memory devices, the performance of computer systems is also limited by latency problems that increase the time required to read data from system memory devices. More

specifically, when a memory device read command is coupled to a system memory device, such as a synchronous DRAM ("SDRAM") device, the read data are output from the SDRAM device only after a delay. Therefore, although SDRAM devices can synchronously output burst data at a high data rate, the delay in initially providing the data or latency can significantly slow the operating speed of a computer system using such SDRAM devices.

Another situation which increases latency in a conventional system memory is where a write command is immediately followed by a read command. When the controller issues a write command, the controller must wait until the write data is no longer present on or has "cleared" the data bus. This waiting by the controller increases the latency of the system memory because the read command cannot be applied to a required memory device until later in time. No data is being transferred on the data bus for a longer time after the write data has cleared the bus due to the latency of the memory devices, which lowers the bandwidth of the system memory. As frequencies increase, conventional system topologies can not meet timing requirements due to physical and electrical limitations. Thus memory hubs, a point to point solution are implemented.

One approach to alleviating the memory latency problem is to use multiple memory devices coupled to the processor through a memory hub. In a memory hub architecture, a system controller or memory controller is coupled over a high speed data link to several memory modules. Typically, the memory modules are coupled in a point-to-point or daisy chain architecture such that the memory modules are connected one to another in series. Thus, the memory controller is coupled to a first memory module over a first high speed data link, with the first memory module connected to a second memory module through a second high speed data link, and the second memory module coupled to a third memory module through a third high speed data link, and so on in a daisy chain fashion.

Each memory module includes a memory hub that is coupled to the corresponding high speed data links and a number of memory devices on the module, with the memory hubs efficiently routing memory requests and responses between the

controller and the memory devices over the high speed data links. Computer systems employing this architecture can have a higher bandwidth because a processor can access one memory device while another memory device is responding to a prior memory access. For example, the processor can output write data to one of the memory devices in the system while another memory device in the system is preparing to provide read data to the processor. Moreover, this architecture also provides for easy expansion of the system memory without concern for degradation in signal quality as more memory modules are added, such as occurs in conventional multi drop bus architectures.

Although computer systems using memory hubs may provide superior performance, they nevertheless may often fail to operate at optimum speeds for a variety of reasons. For example, even though memory hubs can provide computer systems with a greater memory bandwidth, they still suffer from latency problems of the type described above. One problem arises as write commands propagate from one memory hub to another. While a write command is propagating downstream, the controller must wait before issuing a subsequent read command to ensure no collision of data. Thus, although a given write command may be directed to the first hub downstream from the controller, for example, the controller must wait until it is sure the data has propagated to the last hub before issuing a subsequent read command to the last hub. This waiting by the controller delays the issuance of the read command and thereby increases the latency of the memory system.

There is a need for a system and method for reducing the latency of a system memory having a memory hub architecture.

SUMMARY OF THE INVENTION

According to one aspect of the present invention, a memory hub receives downstream memory requests and processes each received downstream memory request to determine whether the memory request includes a write command directed to the memory hub. The memory hub operates in a first mode when the write command is directed to the hub to develop memory access signals adapted to be applied to memory devices. The memory hub operates in a second mode when the write command is not

directed to the hub to provide the command on a downstream output port adapted to be coupled to a downstream memory hub.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system including a system memory having a high bandwidth memory hub architecture according to one example of the present invention.

Figure 2 is a signal timing diagram illustrating the timing in the system memory of Figure 1 when the present write termination scheme is implemented.

Figure 3 is a signal timing diagram illustrating the lower latency of the system memory of Figure 1 executing the write termination scheme of Figure 2 when compared to the timing of a conventional system memory without the write termination scheme.

DETAILED DESCRIPTION OF THE INVENTION

A computer system 100 according to one example of the present invention is shown in Figure 1. The computer system 100 includes a system memory 102 having a memory hub architecture that terminates write data at a destination hub, which allows the controller to issue a read command more quickly after a write command and thereby lowers the latency of the system memory, as will be explained in more detail below. In the following description, certain details are set forth to provide a sufficient understanding of the present invention. One skilled in the art will understand, however, that the invention may be practiced without these particular details. In other instances, well-known circuits, control signals, timing protocols, and/or software operations have not been shown in detail or omitted entirely in order to avoid unnecessarily obscuring the present invention.

The computer system 100 includes a processor 104 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 is typically a central processing unit ("CPU") having a processor bus 106 that normally includes an address bus, a control bus, and a

data bus. The processor bus 106 is typically coupled to cache memory 108, which, as previously mentioned, is usually static random access memory ("SRAM"). Finally, the processor bus 106 is coupled to a system controller 110, which is also sometimes referred to as a "North Bridge" or "memory controller."

The system controller 110 serves as a communications path to the processor 104 for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller 112, which is, in turn, coupled to a video terminal 114. The system controller 110 is also coupled to one or more input devices 118, such as a keyboard or a mouse, to allow an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or retrieve data from internal or external storage media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

The system controller 110 is further coupled to the system memory 102, which includes several memory modules 130a,b...n, and operates to apply commands to the memory modules to optimize the bandwidth of the system memory, as will be discussed in more detail below. The memory modules 130 are coupled in a point-to-point or daisy chain architecture through respective high speed links 134 coupled between the modules and the system controller 110. The high-speed links 134 may be optical, RF, or electrical communications paths, or may be some other suitable type of communications paths, as will be appreciated by those skilled in the art. In the event the high-speed links 134 are implemented as optical communications paths, each optical communication path may be in the form of one or more optical fibers, for example. In such a system, the system controller 110 and the memory modules 130 will each include an optical input/output port or separate input and output ports coupled to the corresponding optical communications paths.

Although the memory modules 130 are shown coupled to the system controller 110 in a daisy architecture, other topologies may also be used, such as a switching topology in which the system controller 110 is selectively coupled to each of the memory modules 130 through a switch (not shown), or a multi-drop architecture in which all of the memory modules 130 are coupled to a single high-speed link 134. Other topologies that may be used, such as a ring topology, will be apparent to those skilled in the art.

Each of the memory modules 130 includes a memory hub 140 for communicating over the corresponding high-speed links 134 and for controlling access to six memory devices 148, which are synchronous dynamic random access memory ("SDRAM") devices in the example of Figure 1. The memory hubs 140 each include input and output ports that are coupled to the corresponding high-speed links 134, with the nature and number of ports depending on the characteristics of the high-speed links. A fewer or greater number of memory devices 148 may, be used, however, and memory devices other than SDRAM devices may also be used. The memory hub 140 is coupled to each of the system memory devices 148 through a bus system 150, which normally includes a control bus, an address bus, and a data bus.

In operation, each memory hub 140 receives downstream memory commands and processes these commands to determine whether a given command is directed to the corresponding memory module 130. More specifically, each memory hub 140 determines whether a given memory command includes a write command. When the memory hub 140 determines a memory request includes a write command, the memory hub next determines whether the write command is directed to the corresponding memory module 130. If this determination is negative, meaning the write command is not directed to the corresponding memory module 130, the memory hub 140 forwards the write command's data to the next downstream memory module. Conversely, if the determination is positive, indicating the write command is directed to the corresponding memory module 130, the memory hub 140 terminates the forwarding of the write command's data to the next downstream memory module. Moreover, the

memory hub may terminate the write command to the next downstream memory module.

Each memory hub 140 thus determines whether a given write command is directed to the corresponding memory module 130, and if directed to that module terminates the propagation of the write command's data to downstream memory modules. Each memory hub 140 also forwards memory responses from downstream memory modules 130 to the next adjacent upstream memory module. Such memory responses may include, for example, read data corresponding to a read command directed to one of the downstream memory modules.

In the following description, a write command or read command is utilized in referring to the actual instruction applied to a memory module 130 and the write or read data associated with the instruction will be referred to separately. A command may be considered, however, to include both the instruction portion and the data portion. Furthermore, it should be noted that each command will include some type of address or identification information which identifies the particular memory module 130 to which the command is directed. The system controller 110 thus accesses a particular memory module 130 by providing identification information in the command for the desired memory module.

The overall operation of the system controller 110 and system memory 102 will now be described in more detail with reference to the signal timing diagram of Figure 2, which illustrates the timing of commands applied to the system memory by the system controller. Figure 2 illustrates an example where the system controller 110 is writing data to the memory module 130a and reading data from the downstream memory module 130b. At a time t1, the system controller 110 applies a read command on the high-speed link 134 and this command is received at the memory module 130a at atime t2.

At this point, the memory hub 140 in the module 130a processes the received command and determines whether the command is directed to that memory module. In the present example, the read command is directed to the memory module 130b and thus the memory hub 140 in the memory module 130a forwards the command

to the memory module 130b where it is received at a time t3. The memory hub 140 in module 130b processes the received read command, determines the command is directed to that module, and thereafter applies the appropriate signals on the bus system 150 to access the desired read data in the memory devices 148. The memory hub 140 in the module 130b places the read data on the high-speed link 134 starting at a time t4, with block representing read data placed on the high speed links 134. The read data is received at the module 130a starting at a time t5 and forwarded upstream to the system controller 110 where it is received starting at a time t 6.

Returning now to the time t 3, while the read command is being received and processed by the memory module 130b the system controller 110 places a write command on the high-speed link 134. The write command includes identification information corresponding to the memory module 130 to which data is to be written, namely module 130a. At a time t 7, the write command is received by the memory hub 140 in the module 130a and processed to determine whether the command is directed to that memory module. In the present example, the memory hub 140 in module 130a determines the write command is directed to that module and thereafter performs two operations. First, the memory hub 140 terminates the write command, meaning that the write command is not provided to the next downstream memory module 130b. Second, the memory hub 140 in module 130a processes the write command and thereafter applies the appropriate signals on the bus system 150 to access the desired storage locations the memory devices 148. In another embodiment of the invention, the write command may be passed to the next downstream hub 140, but the write data may be terminated.

While the memory hub 140 in module 130a is processing the received write command, system controller 110 places write data on the high-speed link 134 starting at a time t 8, where each block once again represents write data placed on the high-speed link. The write data is received at the memory module 130a starting at a time t9, and the memory hub 140 thereafter places the write data on the bus system 150 and develops the appropriate signals to transfer the write data into the desired storage locations in the memory devices 148.

At this point, the system controller 110 has written data to the memory module 130a. Note that the last word of write data being transferred to the module 130a is completed at a time t10, which is just before the time t5 when the first piece of the read data from module 130b is received at the module 130a. Thus, there is no collision of write data in read data on the high-speed link 134 between the system controller 110 and the memory module 130a. Moreover, the read data that is received at the system controller 110 at the time t6 occurs earlier in time than in a conventional system because the memory hub 140 in module 130b need not wait for the write data to pass through that memory module. This is true because the hub 140 in module 130a terminates downstream propagation of the write data once the hub determined the write command was directed to the corresponding module 130a.

In Figure 2, the dotted lines starting at times t11 and t12 represent when the write command and write data, respectively, would have arrived at module 130b if the write data had not been terminated by module 130a. Figure 2 illustrates that if the write data had not been terminated by module 130a, then the read data from module 130b could not be provided until approximately a time t13, which is after the write data passed through module 130b. If it is assumed data is transferred on each edge of clock signal (shown in Figure 2) such that eight data words are transferred in four clock cycles, then termination of the write data results in read data being returned four clock cycles earlier than if the write data was not terminated.

Figure 3 is a signal timing diagram illustrating the lower latency of the system memory 102 of Figure 1 due to the write termination scheme just described with reference to Figure 2 when compared to the timing of a conventional system memory without the write termination scheme. As seen in Figure 3, with the system memory 102 the memory controller 110 issues the read command first at a time t1 and thereafter issues a write command at a time t2. The system controller 110 then places the write data on the high speed link 134 at a time t3 and the read data are received by the controller starting at a time t4 and ending at a time t5. If no write termination was performed by the memory hubs 140, the system controller 110 would not issue the read command until approximately the time t2 and would not receive the read data until a

period starting at a time t6 and ending at a time t7. The time t7 is four clock cycles after the time t5 at which the controller 110 has received all the read data when write termination is performed. Accordingly, the latency of the system memory 102 is reduced by four clock cycles, which is significant since during four clock cycles, 8 data words may be transferred.

The write termination performed by the memory hubs 140, lowers the latency of the system memory 102, and thereby increases the bandwidth of the memory. This write termination scheme is particularly useful when data is to be written to a first memory module 130 that is upstream of a second downstream memory module. As described above, in a conventional system the controller must time the issue of the write command and then time the issuance of the read command so that the write data does not collide with the read data. Termination of the write data at the upstream module 130 allows the controller 110 to actually issue the read command before the write command, which is the converse of a conventional system, and the earlier issuance of the read command lowering the latency of the system memory.

With the system memory 102, the system controller 110 knows the physical location of the modules 130 relative to one another, and thus knows precisely when to issue the read and write commands. For example, data is to be read from memory module 130z and written to memory module 130a, the controller 110 may issue the read command very early relative to the write command since the returning read data will be delayed as it progresses through the intervening hubs 140 on the corresponding memory modules 130b-y. In contrast, if data is to be read from memory module 130b and written to memory module 130a, the controller 110 will still issue the read command prior to the write command but not as early as in the prior situation where data was being read from module 130z.

One skilled in the art will understand suitable circuitry for forming the components of the computer system 100, such as the memory hubs 140 so that these components perform the described functionality. In the preceding description, certain details were set forth to provide a sufficient understanding of the present invention. One skilled in the art will appreciate, however, that the invention may be practiced

without these particular details. Furthermore, one skilled in the art will appreciate that the example embodiments described above do not limit the scope of the present invention, and will also understand that various equivalent embodiments or combinations of the disclosed example embodiments are within the scope of the present invention. Illustrative examples set forth above are intended only to further illustrate certain details of the various embodiments, and should not be interpreted as limiting the scope of the present invention. Also, in the description above the operation of well known components has not been shown or described in detail to avoid unnecessarily obscuring the present invention. Finally, the invention is to be limited only by the appended claims, and is not limited to the described examples or embodiments of the invention.

CLAIMS

1. A method of processing write commands in a memory system having a memory hub architecture including a plurality of memory hubs coupled in a point-to-point architecture, the method comprising:

applying a write command to a first hub in the system;

determining in the first hub whether the write command is directed to that hub;

when the determination indicates the write command is directed to the first hub, terminating downstream forwarding of the write data;

when the determination indicates the write command is not directed to the first hub, forwarding the write data downstream to a second hub; and

repeating the operations of determining through when the determination indicates the write command is not directed for all required memory hubs.

2. The method of claim 1 wherein the memory system includes a series of memory hubs in addition to the first and second hubs, and wherein the method further comprises repeating the acts of claim 1 except for the act of applying a write command to a first hub in the system for each of the memory hubs in the series.

3. The method of claim 2, further comprising forwarding the write command beyond a memory hub in which downstream forwarding of the write data is terminated until the write command reaches a final memory hub in the series of memory hubs.

4. The method of claim 1 wherein the write command includes a command portion and a write data portion, and wherein terminating downstream forwarding of the command comprises terminating the forwarding of both the command portion and write data portion.

5. The method of claim 1 further comprising applying a read command to the first hub prior to applying the write command to the first hub, the read command being directed to a hub that is downstream of the hub to which the write command is directed.

6. The method of claim 5 wherein a time interval is defined between when the read and write commands are applied, and wherein the time interval is a function of the position of the downstream hub to which the read command is directed relative to the upstream hub to which the write command is directed.

7. A method of operating a system memory having a memory hub architecture, the system memory including a plurality of memory modules coupled in series, each memory module including a memory hub, and the method comprising:

detecting in each memory hub whether a write command is directed to the corresponding memory module;

when the operation of detecting indicates the write command is directed to the corresponding module, terminating the forwarding of the write data to downstream memory modules.

8. The method of claim 7 wherein detecting comprises comparing an address component of the write command to an address associated the memory module.

9. The method of claim 7 wherein the write command includes a command portion and a write data portion, and wherein terminating the forwarding of the write data to downstream memory modules comprises terminating the forwarding of write data portion.

10. The method of claim 7 further comprising applying a read command to a first memory module prior to the write command.

11. The method of claim 10 wherein a time interval is defined between when the read and write commands are applied, and wherein the time interval is a function of the position of the memory module to which the read command is directed relative to the memory module to which the write command is directed.

12. The memory hub of claim 11 wherein the hub is operable in the second mode to provide received write data on a downstream data port adapted to be coupled to a downstream memory hub.

13. The memory hub of claim 12 wherein the memory hub is operable in the first mode to provide the command on the downstream output port and to terminate providing received write data on the downstream data port.

14. A memory hub adapted to receive downstream memory requests and operable to process each received downstream memory request to determine whether the memory request includes a write command directed to the memory hub, and the memory hub operable in a first mode when the write command is directed to the hub to develop memory access signals adapted to be applied to memory devices, and the hub operable in a second mode when the write command is not directed to the hub to provide the command on a downstream output port adapted to be coupled to a downstream memory hub.

15. The memory hub of claim 14 wherein the write command comprises a command portion and a data portion.

16. The memory hub of claim 14 wherein the downstream memory request includes an address portion, and wherein the memory hub determines whether the write command is directed to the memory hub comprises comparing a value of the address portion to an address of the memory hub.

17. The memory hub of claim 14 wherein the memory access signals comprise address, control, and data signals to be applied to memory devices.

18. The memory hub of claim 14 wherein the hub is further adapted to receive upstream memory responses and provide such responses on an upstream output port adapted to be coupled to an upstream memory hub.

19. The memory hub of claim 14 wherein the downstream output port comprises an optical port.

20. A memory module, comprising:

a plurality of memory devices; and

a memory hub coupled to the memory devices and including a downstream input port adapted to receive downstream memory requests, and the hub operable to process each received downstream memory request to determine whether the memory request includes a write command directed to the memory module, and the memory hub operable in a first mode when the write command is directed to the module to apply memory access signals to the memory devices, and the hub operable in a second mode when the write command is not directed to the module to provide the command on a downstream output port.

21. The memory module of claim 20 wherein the memory devices comprise dynamic random access memory devices.

22. The memory module of claim 20 wherein the hub is operable in the second mode to provide received write data on a downstream data port.

23. The memory module of claim 22 wherein the hub is operable in the first mode to provide the command on the downstream output port and to terminate providing received write data on the downstream data port.

24. The memory module of claim 20 wherein the write command comprises a command portion and a data portion.

25. The memory module of claim 20 wherein the downstream memory request includes an address portion, and wherein the memory hub determines whether the write command is directed to the memory hub comprises comparing a value of the address portion to an address of the memory hub.

26. The memory module of claim 20 wherein the memory access signals comprise address, control, and data signals to be applied to memory devices.

27. The memory hub of claim 20 wherein the hub is further adapted to receive upstream memory responses and provide such responses on an upstream output port adapted to be coupled to an upstream memory hub.

28. A memory system, comprising:

a system controller; and

a plurality of memory modules, each memory module being coupled to adjacent memory modules through respective high-speed communications links, at least one of the memory modules being coupled to the system controller through a respective high-speed communications link, and each memory module comprising:

a plurality of memory devices; and

a memory hub coupled to the memory devices and including a downstream input port adapted to receive downstream memory requests, and the hub operable to process each received downstream memory request to determine whether the memory request includes a write command directed to the memory module, and the memory hub operable in a first mode when the write command is directed to the module to apply memory access signals to the memory devices, and the hub operable in a second mode when the write command is not directed to the module to provide the command on a downstream output port.

29. The memory system of claim 28 wherein the system controller comprises a memory controller.

30. The memory system of claim 28 wherein the system controller determines a timing between when the controller applies a read command to a first memory module relative to when the controller applies a write command to a second module that is upstream of the first module as a function of the position of the downstream module to which the read command is directed relative to the upstream module to which the write command is directed.

31. The memory system of claim 30 wherein the time between when the read and write commands increase the farther downstream is the downstream module relative to the upstream module.

32. The memory system of claim 28 wherein the memory devices comprise dynamic random access memory devices.

33. The memory system of claim 28 wherein the write command comprises a command portion and a data portion.

34. The memory system of claim 28 wherein the downstream memory request includes an address portion, and wherein each memory hub determines whether the write command is directed to the memory by comparing a value of the address portion to an address associated with the memory module.

35. The memory system of claim 28 wherein the memory access signals comprise address, control, and data signals.

36. The memory system of claim 28 wherein each memory hub receives upstream memory responses from an adjacent downstream module and provides such responses to an adjacent upstream memory module.

37. A computer system, comprising:

a processor;

a system controller coupled to the processor;

an input device coupled to the processor through the system controller;

an output device coupled to the processor through the system controller;

a storage device coupled to the processor through the system controller; and

a plurality of memory modules coupled to the system controller, each memory module being coupled to adjacent memory modules through respective high-speed communications links, at least one of the memory modules being coupled to the system controller through a respective high-speed communications link, and each memory module comprising:

a plurality of memory devices; and

a memory hub coupled to the memory devices and including a downstream input port adapted to receive downstream memory requests, and the hub operable to process each received downstream memory request to determine whether the memory request includes a write command directed to the memory module, and the memory hub operable in a first mode when the write command is directed to the module to apply memory access signals to the memory devices, and the hub operable in a second mode when the write command is not directed to the module to provide the command on a downstream output port.

38. The computer system of claim 37 wherein each high-speed communications link comprises an optical link.

39. The computer system of claim 37 wherein the system controller determines a timing between when the controller applies a read command to a first memory

module relative to when the controller applies a write command to a second module that is upstream of the first module as a function of the position of the downstream module to which the read command is directed relative to the upstream module to which the write command is directed.

40. The computer system of claim 39 wherein the time between when the read and write commands increase the farther downstream is the downstream module relative to the upstream module.

41. The computer system of claim 37 wherein the memory devices comprise dynamic random access memory devices.

42. The computer system of claim 37 wherein the write command comprises a command portion and a data portion.

43. The computer system of claim 37 wherein the downstream memory request includes an address portion, and wherein each memory hub determines whether the write command is directed to the memory by comparing a value of the address portion to an address associated with the memory module.

44. The computer system of claim 37 wherein each memory hub receives upstream memory responses from an adjacent downstream module and provides such responses to an adjacent upstream memory module.

45. The computer system of claim 37 wherein the processor comprises a central processing unit ("CPU").

METHOD AND SYSTEM FOR TERMINATING WRITE COMMANDS IN A HUB-BASED MEMORY SYSTEM

ABSTRACT OF THE DISCLOSURE

A memory hub receives downstream memory commands and processes each received downstream memory command to determine whether the memory command includes a write command directed to the memory hub. The memory hub operates in a first mode when the write command is directed to the hub to develop memory access signals adapted to be applied to memory devices. The memory hub operates in a second mode when the write command is not directed to the hub to provide the command's write data on a downstream output port adapted to be coupled to a downstream memory hub.

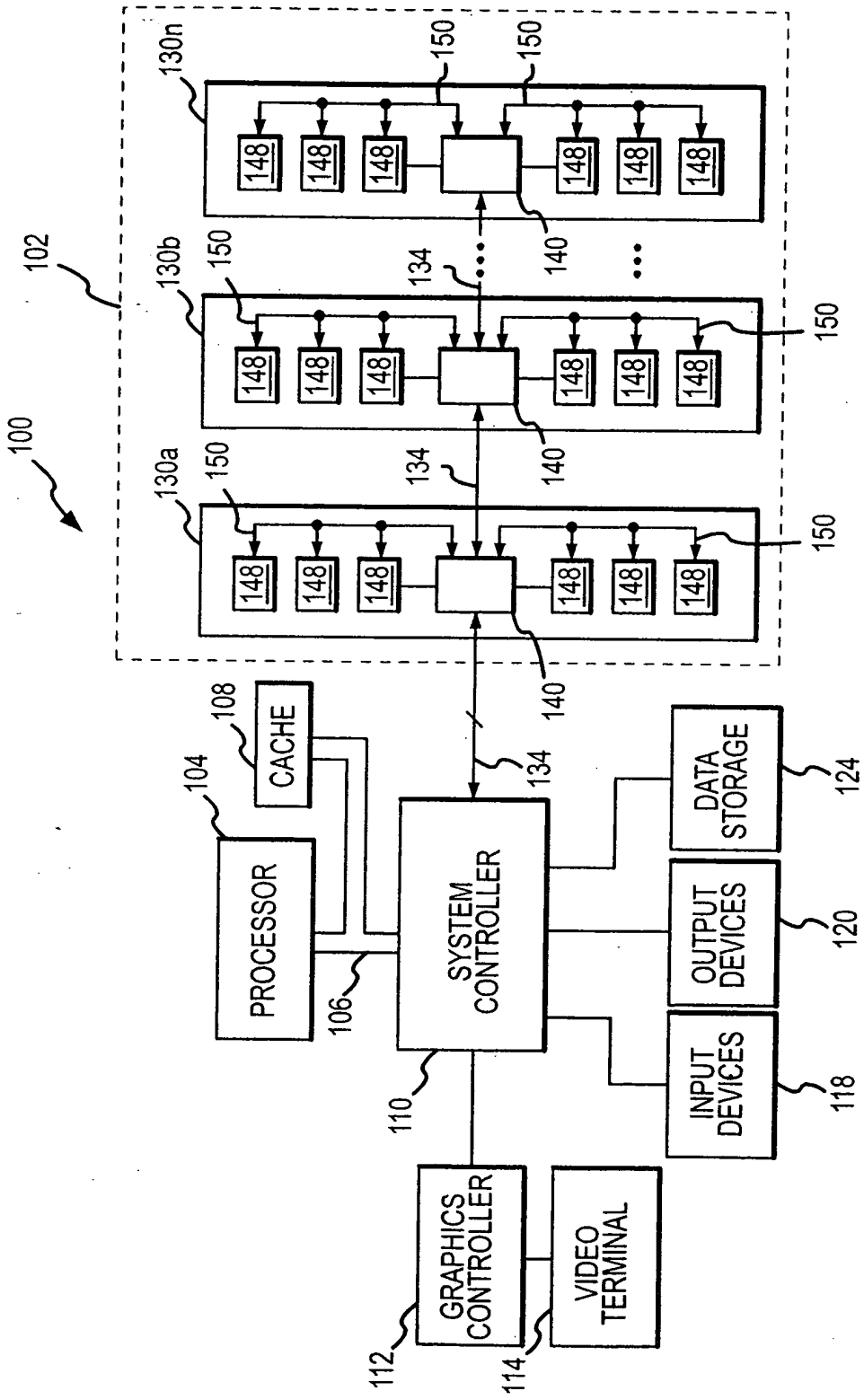
100
102
130n
130b
130a
150
148
140
134
104
PROCESSOR
108
CACHE
106
SYSTEM CONTROLLER
110
DATA STORAGE
124
OUTPUT DEVICES
120
INPUT DEVICES
118
GRAPHICS CONTROLLER
112
VIDEO TERMINAL
114

FIG.1

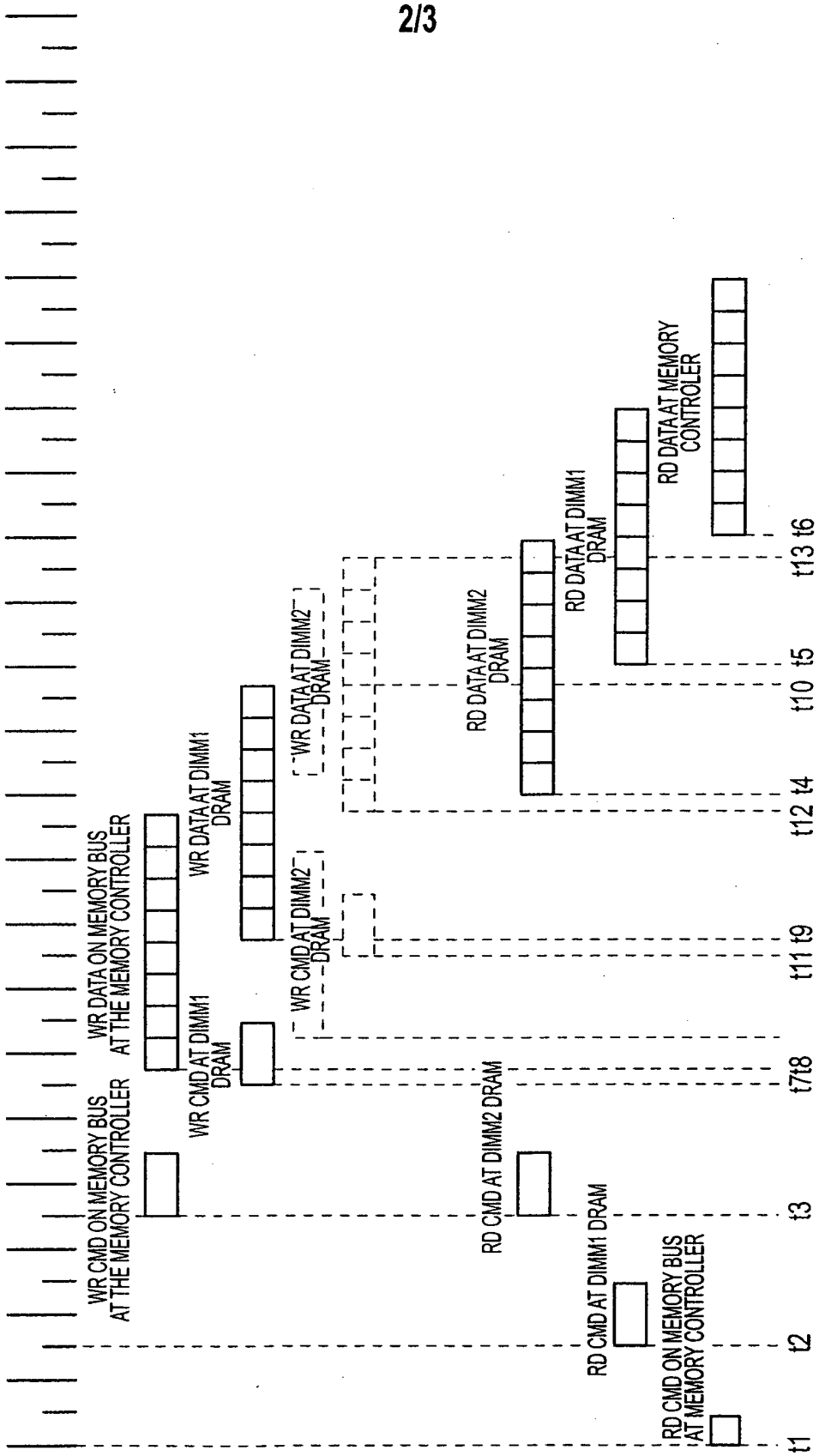
WR CMD ON MEMORY BUS AT THE MEMORY CONTROLLER
WR DATA ON MEMORY BUS AT THE MEMORY CONTROLLER
WR CMD AT DIMM1 DRAM
WR DATA AT DIMM1 DRAM
WR CMD AT DIMM2 DRAM
WR DATA AT DIMM2 DRAM
RD CMD AT DIMM2 DRAM
RD DATA AT DIMM2 DRAM
RD CMD AT DIMM1 DRAM
RD DATA AT DIMM1 DRAM
RD CMD ON MEMORY BUS AT MEMORY CONTROLLER
RD DATA AT MEMORY CONTROLLER
t1
t2
t3
t7t8
t11 t9
t12 t4
t10 t5
t13 t6

FIG.2

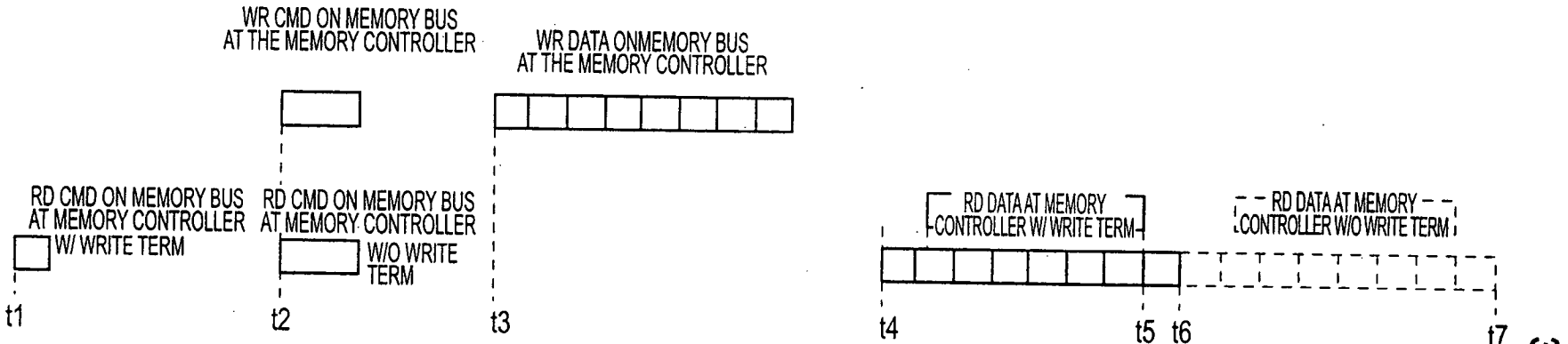
WR CMD ON MEMORY BUS
AT THE MEMORY CONTROLLER
WR DATA ONMEMORY BUS
AT THE MEMORY CONTROLLER
RD CMD ON MEMORY BUS
AT MEMORY CONTROLLER
W/ WRITE TERM
RD CMD ON MEMORY BUS
AT MEMORY CONTROLLER
W/O WRITE
TERM
RD DATA AT MEMORY
CONTROLLER W/ WRITE TERM
RD DATA AT MEMORY
CONTROLLER W/O WRITE TERM
t1
t2
t3
t4
t5
t6
t7

FIG.3